Proposal

Project Proposal: Customer Churn Prediction

1. Introduction

Customer churn, the phenomenon where customers cease doing business with a company, is a critical problem across many industries, particularly subscription-based services like telecommunications, streaming platforms, and SaaS. It is a well-established fact that acquiring a new customer is significantly more expensive than retaining an existing one.

Therefore, proactively identifying customers who are at a high risk of churning is a key business priority. This project proposes the development of a machine learning model to predict customer churn based on their demographic data and service usage patterns. By accurately forecasting churn, a business can implement targeted retention strategies, such as special offers or proactive support, to save valuable customers and reduce revenue loss.

---

2. Problem Statement

A company (e.g., a telecom or utility provider) is experiencing significant customer churn but lacks an effective way to identify *which* customers are at risk of leaving. This leads to two primary issues:

1. **Revenue Loss:** High-value customers are lost because the company cannot intervene in time.
2. **Inefficient Retention Spending:** Retention efforts (like discounts) are either applied too broadly (costing too much) or not at all, rather than being focused on the small group of customers who are actually at risk.

The core problem is the lack of an accurate, data-driven system to forecast churn risk for individual customers, preventing timely and targeted retention strategies. This project

aims to solve this by building a predictive classification model.

---

3. Objectives

The primary goal of this project is to build and evaluate a robust machine learning model that accurately predicts customer churn.

Specific objectives include:

- **Data Analysis:** To perform a comprehensive Exploratory Data Analysis (EDA) on the Customer Churn dataset to identify key factors and patterns associated with churn.
- **Data Preprocessing:** To clean the data, handle any missing values, and encode categorical features into a machine-readable format.
- **Model Implementation:** To implement and evaluate several classification algorithms (e.g., Logistic Regression, K-Nearest Neighbors, SVM, Decision Tree, and Random Forest).
- **Handle Class Imbalance:** To address the significant class imbalance (88% Churn vs 12% No-Churn) using **Strategic Undersampling**. We will sample the majority class to achieve a realistic 2:1 ratio, allowing the model to learn the minority class effectively without the noise sometimes introduced by synthetic data generation.
- **Optimization:** To tune the models using GridSearchCV to find the optimal hyperparameters for the best-performing algorithm.
- **Evaluation:** To select the best model based on a comprehensive evaluation, prioritizing metrics like **Recall** and **F1-Score** for the minority class and overall **Accuracy**.
- **Deliverable:** To save the final, trained model and the data scaler so they can be used in a practical application (like the provided app.py).

---

4. Related Works

Customer Churn Prediction is a well-established field in data mining. A review of existing literature shows that various models have been successfully applied:

- **Baseline Models:** Logistic Regression is frequently used as a baseline model due to its simplicity and high interpretability, providing a benchmark for other models.
- **Ensemble Methods:** Tree-based ensemble models, particularly **Random Forest** and **Gradient Boosting (e.g., XGBoost)**, are consistently reported as top performers. Their ability to handle non-linear relationships and their robustness against overfitting (in the case of Random Forest) make them highly effective for this problem.
- **Other Classifiers:** K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) are also widely used, with performance often depending on careful data scaling and hyperparameter tuning.
- **Handling Imbalance:** A recurring theme in churn-related work is the challenge of class imbalance. While many studies use SMOTE, this project takes a data-centric approach by employing **Undersampling**. Given the extreme "inverted" imbalance of our specific dataset, this method ensures the model is trained on a cleaner, more representative distribution of classes.

---

5. Methodology

The project follows a structured machine learning pipeline:

1. **Data Collection & Loading:** Load the customer churn dataset from CSV.
2. **Exploratory Data Analysis (EDA):** Analyze feature distributions, check for missing values, and understand correlations.
3. **Data Preprocessing:**
   - Handle missing values (e.g., fill InternetService NaN with "None")
   - Encode categorical variables (Label Encoding for binary, mapping for multi-class)
   - Feature selection based on domain knowledge and correlation analysis
4. **Handling Class Imbalance:** Apply strategic undersampling to balance the dataset.
5. **Feature Scaling:** Standardize numerical features using StandardScaler.
6. **Model Training:** Train multiple classifiers (Logistic Regression, KNN, SVM, Random Forest).
7. **Hyperparameter Tuning:** Use GridSearchCV for optimal parameter selection.
8. **Model Evaluation:** Compare models using accuracy, precision, recall, and F1-score.
9. **Model Export:** Save the best model and scaler for deployment.

---

6. Expected Outcomes

- A trained machine learning model with high accuracy in predicting customer churn
- A web-based application (Streamlit) for real-time churn prediction
- Comprehensive documentation including EDA insights and model performance metrics
- Reusable codebase following best practices for reproducibility